Eur päisches Patentamt
Eur pean Patent Office
Office européen des brevets

Bescheinigung Certificate Attestation

Die angehefteten Unterlagen stimmen mit der ursprünglich eingereichten Fassung der auf dem nächsten Blatt bezeichneten europäischen Patentanmeldung überein.

The attached documents are exact copies of the European patent application described on the following page, as originally filed.

Les documents fixés à cette attestation sont conformes à la version initialement déposée de la demande de brevet européen spécifiée à la page suivante.

Patentanmeldung Nr. Patent application No. Demande de brevet n°

01100246.6

Der Präsident des Europäischen Patentamts;
Im Auftrag

For the President of the European Patent Office

Le Président de l'Office européen des brevets
p.o.

I.L.C. HATTEN-HECKMAN

DEN HAAG, DEN
THE HAGUE, 13/08/01
LA HAYE, LE

Eur päisches Patentamt
Eur pean Patent Office
Office européen des brevets

Blatt 2 der Bescheinigung
Sheet 2 of the certificate
Page 2 de l'attestation

Anmeldung Nr.:
Application no.: 01100246.6
Demande n°:

Anmeldetag:
Date of filing: 03/01/01
Date de dépôt:

Anmelder:
Applicant(s):
Demandeur(s):
International Business Machines Corporation
Armond, NY 10504
UNITED STATES OF AMERICA

Bezeichnung der Erfindung:
Title of the invention:
Titre de l'invention:
Method for the synchronization of data

In Anspruch genommene Priorität(en) / Priority(ies) claimed / Priorité(s) revendiquée(s)

Staat:	Tag:	Aktenzeichen:
State:	Date:	File no.
Pays:	Date:	Numéro de dépôt:

Internationale Patentklassifikation:
International Patent classification:
Classification internationale des brevets:
/

Am Anmeldetag benannte Vertragstaaten:
Contracting states designated at date of filing: AT/BE/CH/CY/DE/DK/ES/FI/FR/GB/GR/IE/IT/LI/LU/MC/NL/PT/SE/TR
Etats contractants désignés lors du depôt:

Bemerkungen:
Remarks:
Remarques:

D E S C R I P T I O N

Method for the Synchronization of Data

Field of the Invention

The present invention in general relates to the sharing of data between a plurality of databases. More specifically, the invention relates to the synchronization of such data. Still more specifically, the invention pertains to the synchronization of data between a server system and a mobile client.

Background of the Invention

The rapid acceptance of computer technology by corporations as well as consumers has led to the widespread use of computers. Further abetting this process is the continual reduction in size and cost of personal computers. Originally, personal computers were large items best suited for floor standing or desktop use. Eventually, they became smaller so that desktop use became the standard. Improvements in processor, memory and data storage capabilities have resulted in light weight, powerful mobile computers such as portables, luggables, laptops, notebooks, palm top and personal digital assistants. These computers can provide sufficient processing capability for audio visual applications, such as computer aided design, three dimensional animation, and multimedia presentation, even when used at remote locations.

Typically, many users of such mobile computers also own desktop computers used for applications that manage databases similar to the databases carried in mobile computers. Normally, the user adds data to the two computers independently, e.g., one enters data into the mobile computer, often referred to as a client, when out at a customer site but enters data into the desktop

computer or a network server when in the office. In such cases, the user normally would at some point want to keep these two sets of data identical, especially if different changes are made in parallel to both devices. This action is called data synchronization.

Today, a client needs a quite large piece of software installed to retrieve and add data during the synchronization. The data is exchanged in a format that the client and server are both able to generate and to parse. The client needs to create and to parse the document which requires a certain amount of code on the client device. Especially with today's mobile clients, like mobile phones or small handheld computers, the resources available are very limited. However, adding more flash memory to the mobile clients is still very costly.

A client that wants to synchronize its data with the data of a server needs a local client sync engine that retrieves the adds, changes, updates, etc. from the client's database and creates a standardized document to tell the server which data has been added, deleted, etc., on the client's side. The server parses this document and updates the host database accordingly. The server also checks the host database for changes which might have taken place since the last synchronization cycle with the respective client, retrieves these records and builds the exchange document. The client will receive this document and now needs to parse it to know what to do.

Thus today, a client needs the following three components shown in Fig. 1 to synchronize data with a server: a client database adapter, a sync client agent and a client sync adapter. The sync adapter contains an encoder and a parser to generate and understand the standardized exchange document.

One of the Industry Data Synchronization Standards is SyncML. SyncML includes a universal data synchronization format that is defined by an Extensible Markup Language (XML) document type, XML

being a standard format for interchanging structured documents. This format is exchanged as SyncML messages between network devices. A SyncML message is just an XML document. SyncML is independent from the underlying transport layer and can be used in wireless as well as in wired environments.

Unlike HTML, XML enables the creation of user defined tags. Traditional HTML tags are extended by user defined elements. Like in HTML, the start and end of each logical block is marked by the appropriate tags. Document Type Definitions (DTD) are an important element of XML. They define the role of text elements in a formal model. The DTD can be used to check if an XML document contains valid tags and if the tags occur in the right place within the document. It also specifies the attributes that belong to an element and the valid values of these attributes. Thus, DTDs have two main functions:

a) they specify which document structures can be used by the author of an XML document and have to be handled by the generator that creates that defined kind of XML document; and

b) they specify which document structures have to be handled by parsers that process that defined kind of XML document.

In the Standard Generalized Markup Language (SGML), of which XML is a subset, the use of DTDs is required. However, XML does not require the use of a DTD. A parser that processes a document without a DTD has to extract the relevant information from the document itself.

The statement that XML is „extensible" can be misunderstood. XML defines a syntax, in the form of a number of rules in order to define document structures. These rules are defined in the XML specification and cannot be extended. „Extensible" means that special instances of tag languages can be built on the fundamental rules of XML.

The basic concept of XML is the composition of documents using a series of entities. The entities themselves are composed of one or more logical elements. The elements can contain attributes which describe the way in which the elements have to be processed.

In addition, SyncML defines a synchronization protocol. This protocol specifies how SyncML conformant messages are exchanged to synchronize databases on different network devices. The synchronization protocol supports both one-way, as well as two-way data synchronization. The SyncML specifications also define HTTP, OBEX, and WSP transport bindings, which describe the minimum set of features that a SyncML compliant transport implementation must support.

US-A-5,974,238 discloses an apparatus for performing dynamic synchronization between data stored in a handheld computer and a host computer, each having a plurality of data sets including at least one common data set, each computer having a copy of the common data set. The handheld computer comprises a data synchronization engine, having a pseudo-cache and one or more tags connected to the pseudo cache. Data is synchronized whenever data is written to main memory and/or when the associated pseudo-cache tag is invalidated. By strict adherence to a set of protocols, data coherency is achieved because the system always knows who owns the data, who has a copy of the data, and who has modified the data. The data synchronization engine resolves any differences in the copies and allows the storage of identical copies of the common data set in the host computer and in the handheld computer.

In US-A-5,684,990, a data processing method for synchronizing the data records of a plurality of disparate databases is disclosed, in which a status file is provided containing data records representative of the contents of data records existing in the disparate databases at a prior synchronization. Data records from at least a first and a second of the plurality of databases are

compared to corresponding data records of the status file to determine whether data records of the plurality of databases have changed or been deleted since the prior synchronization, or whether there are new data records since the earlier synchronization. Based on the outcome of the comparing step, decisions are made as to how to update the data records of the first and second databases. Finally, the status file is updated so that its data records are representative of the contents of the data records of the first and second databases after they have been updated.

The known synchronization systems, however, have the big disadvantage that they do not take into account that mobile clients like mobile phones, etc., only have restricted storage capacity. The components necessary for the systems according to the state of the art, however, require a lot of storage capacity on the side of the client device.

In addition, a fixed protocol and document format is used that the client needs to convert the data to and from.

Summary of the Invention

It is therefore an object of the present invention to provide a method for synchronizing data between a server and a client that allows to reduce the amount of code needed on a client to synchronize data with a server system.

It is a further object of the invention to provide such a method that will allow to send data from the client to the server system in a format most suitable for the client device.

It is still another object of the present invention to provide a method for synchronizing data that will free the client device from having to parse the exchange document.

These and other objects and advantages will be achieved by the method disclosed in claim 1 and the apparatus disclosed in claim 10.

Advantageous embodiments of the invention are disclosed in the dependent claims.

Brief Description of the Drawings

The invention will in the following be described in more detail in conjunction with the drawings, in which

Fig. 1 schematically depicts the synchronization architecture according to the state of the art; and

Fig. 2 schematically depicts the synchronization flow according to the invention.

Detailed Description of the Preferred Embodiment

In order to reduce the amount of code needed on a client to synchronize data with a respective server system, some setup is done at the time the server and the client first synchronize a certain database, to enable the server to understand the data from the client and to know how to build the client commands in the right way.

Therefore, the client tells the server where to find in the data block the information the server needs. In case of the database being, e.g., an address book, this would be, e.g., name, street, ZIP-code, city, etc., as, e.g., defined in the vcard specification.

This can be done, e.g., in the following form using a header information:

<Internal structure of the document>, e.g., fixed records, TLV,..
total block length: 80
name: 1-19
street: 20-44
city: 45-59
...

In addition to that, the header information may include some information as the kind of client that has sent the respective information.

The client sends information to the server to enable the server to identify the respective client and to select the correct method to generate the respective client commands.

In an other embodiment, the client sends information to the server as to how to generate the correct client commands in a way that the client just needs to execute the commands the server sends. Each command, like „ADD", „DELETE", „UPDATE", ... has to be defined in this document sent to the server.

After this setup is done, the actual synchronization can take place. In case of a client initiated synchronization, the following steps will be carried out. It has to be noted that the method according to the invention will as well be operable in case of a server initiated synchronization, thereby using protocols that are technically able to carry out such a server initiated synchronization, like, e.g., TCP/IP or the like.

However, in the following only the client initiated synchronization will be described. The steps of this method are

a) The client identifies the changed records;

b) The client dumps the respective records as exactly as stored in the client's database without any convertion in a "standard" format. The client sends the data blocks for

changed, updated, deleted items;

c) The server retrieves the data changed in the server database since the last synchronization cycle;

d) The server interprets the received data based on the information he received during setup;

e) The server resolves any possible conflicts and other needed things, e.g., ID mapping, etc.;

f) The server takes the required actions to update the host database with the information received from the client;

g) The server generates the client commands based on the description received from the client or by knowing the client device; and

h) The client executes the received commands.

The commands which the server sends to the client to execute could be in a form which requires some interpreter on the client, or ideally may be already object code compiled by the server for execution on the client. In this case, the client would just need a very small daemon or the like which downloads and starts the precompiled object code.

The size of the data to be transferred between the client and the server can be slightly larger than with known methods, but within the next two years the datarate of wireless networks will increase explosively as soon as, e.g., UMTS is introduced. In parallel, the processor power of client devices will increase moderately, therefore transferring data over wireless networks will be much cheaper in comparison with the costs of memory and processing power in client devices.

The proposed invention allows reducing the required software in

the client extremely. This makes it possible to produce cheaper clients with less memory, or to use the memory available for other additional applications.

Performance and memory on the server is usually much cheaper as on the client. The method according to the invention therefore moves the workload from the client to the server and thus makes the complete solution more cost effective.

C L A I M S

1. Method for synchronizing data records being stored in a database of a server system as well as in a database on a client, said synchronization being initiated by said client, comprising the steps of:

   a) providing a setup information by said client representing the data format of the data present on said client as well as a list of commands being executable on said client;

   b) detecting the changed records in said client's database;

   c) dumping said detected records without any conversion;

   d) sending said dump to said server system;

   e) retrieving the records changed in said server system's database;

   f) interpreting data present in said changed records on the basis of said setup information;

   g) updating said server system's database;

   h) compiling a program by said server system to update said client's database based on said setup information;

   i) sending said program to said client; and

   j) executing said program by said client to thereby update said client's database.

2. Method according to claim 1, wherein said client is a mobile client.

3. Method according to claim 2, wherein said mobile client is a mobile phone, a handheld computer, or a PDA.

4. Method according to claim 1, additionally comprising the step of resolving any possible conflicts by the server system between steps f) and g).

5. Method according to any one of claims 1 to 3, wherein said setup information comprises a header portion.

6. Method according to claim 5, wherein said server interprets the data sent by the client based on said header portion.

7. Method according to any one of the preceding claims, wherein said program is executed by said client using an interpreter.

8. Method according to any one of claims 1 to 4, wherein said program comprises object code compiled by the server system.

9. Method for synchronizing data records being stored in a database of a server system as well as in a database on a client, said synchronization being initiated by said server system, comprising the steps of:

a) providing a setup information by said client representing the data format of the data present on said client as well as a list of commands being executable on said client;

b) retrieving the records changed in said server system's database;

c) compiling a program by said server system to update said client's database based on said setup information;

d) sending said program to said client; and

e) executing said program by said client to thereby update said client's database.

10. Method according to claim 9, wherein said setup information comprises information to enable the server to identify the respective client and to select the correct method to generate the respective client commands.

11. A data processing system for synchronizing data records stored on a database of a server system and on a database of a client, said client comprising means for dumping said database and a command executer.

12. The system of claim 7, wherein said client additionally comprises means for downloading and starting precompiled object code.

13. The system of claim 7, wherein said server system additionally comprises a plurality of compilers.

14. A computer program product stored on a computer usable medium comprising computer readable program means for cousing a computer to perform the method of anyone of the claims 1 to 10 when said program is executed on said computer.

A B S T R A C T

A method for synchronizing data records being stored in a database of a server system as well as in a database on a client is provided which allows reducing the required software in the client extremely. This makes it possible to produce cheaper clients with less memory, or to use the memory available for other additional applications.
(Fig. 2)

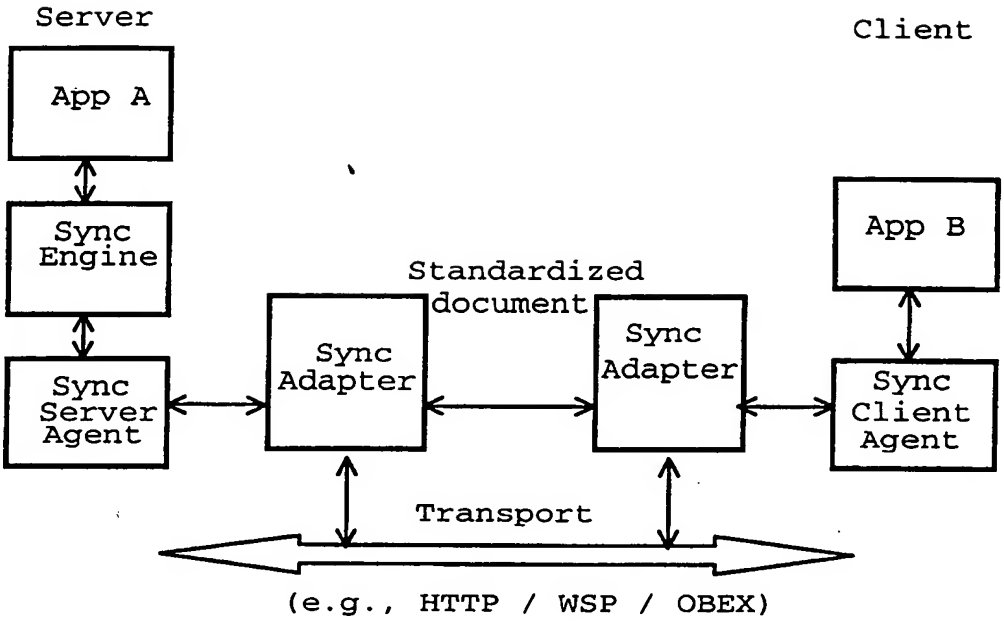

FIG. 1

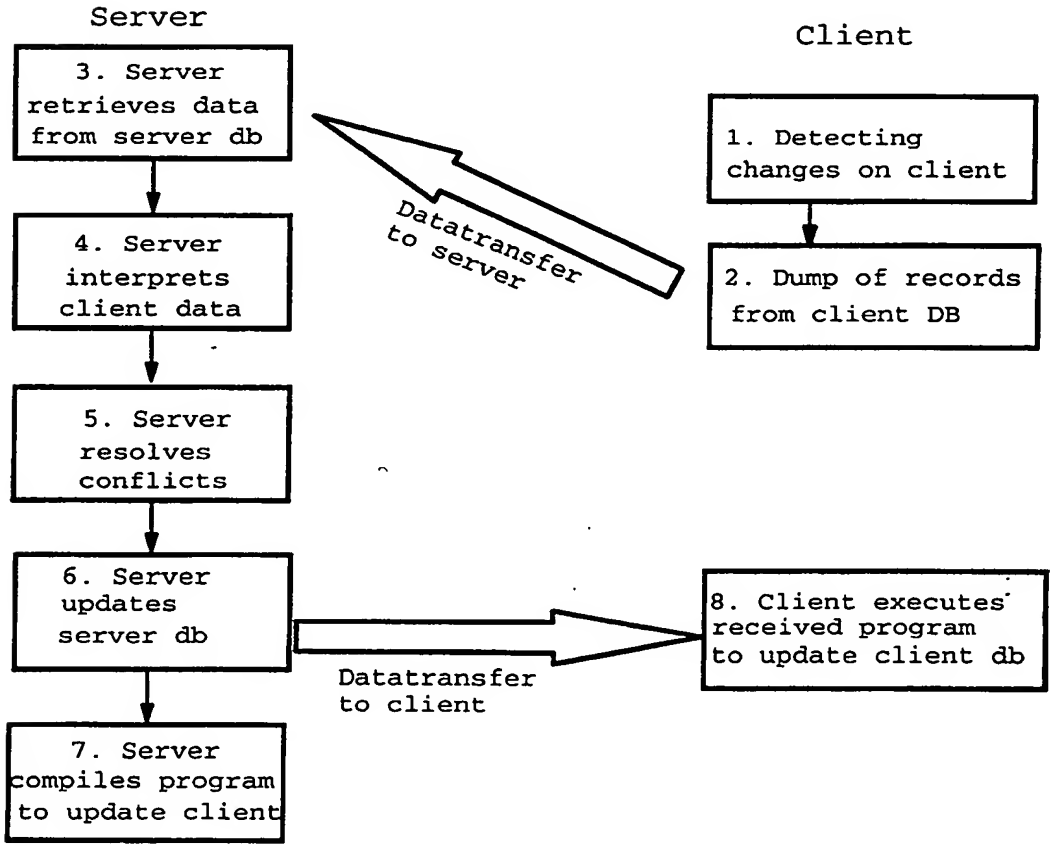

FIG. 2

THIS PAGE BLANK (USPTO)